

WHAT IS CLAIMED IS:

1 1. A method of maintaining an index during a reorganization of data in a
2 database, wherein said database comprises one or more records and each said record
3 includes a root segment, the method comprising:
4 storing each root segment in a separate storage location; and
5 retaining each root segment in its storage location during a reorganization of
6 data in said database.

1 2. The method of claim 1, wherein all root segments stored within a block
2 of storage locations are stored in contiguous storage locations.

1 3. The method of claim 1, wherein each said storage location which
2 stores a root segment is fixed.

1 4. The method of claim 2, wherein each said storage location which
2 stores a root segment is fixed.

1 5. The method of claim 1, wherein said database is an IMS full function
2 database.

1 6. The method of claim 3, wherein each said root segment is stored in a
2 fixed storage location at the time it is added to the database.

1 7. A method of maintaining an index during a reorganization of data in a
2 database, wherein said database comprises one or more records and each said record
3 includes a root segment and one or more non-root segments, the method comprising:
4 storing each root segment in a separate storage location, wherein each storage
5 location is associated with a particular block of storage locations;
6 retaining each root segment in its storage location during a reorganization of
7 data in said database; and
8 storing each non-root segment, associated with a first root segment, in a block
9 of storage locations in which said first root segment is also stored.

1 8. The method of claim 7, wherein all root segments stored within a block
2 of storage locations are stored in contiguous storage locations.

1 9. The method of claim 7, wherein each said storage location which
2 stores a root segment is fixed.

1 10. The method of claim 8, wherein each said storage location which
2 stores a root segment is fixed.

1 11. The method of claim 7, wherein said database is an IMS full function
2 database.

1 12. The method of claim 9, wherein each said root segment is stored in a
2 fixed storage location at the time it is added to the database.

1 13. A method of maintaining an index during a reorganization of data in a
2 database, wherein said database comprises one or more records, each said record
3 includes a root segment and one or more non-root segments, and each root segment
4 comprises a prefix component and a data component, the method comprising:
5 storing said prefix component of each root segment in a separate storage
6 location; and
7 retaining said prefix component of each root segment in its storage location
8 during a reorganization of data in said database.

1 14. The method of claim 13, wherein all root segment prefix components
2 stored within a block of storage locations are stored in contiguous storage locations.

1 15. The method of claim 13, wherein each said storage location which
2 stores a root segment prefix component is fixed.

1 16. The method of claim 14, wherein each said storage location which
2 stores a root segment prefix component is fixed.

1 17. The method of claim 13, wherein said database is an IMS full function
2 database.

1 18. The method of claim 15, wherein each said root segment prefix
2 component is stored in a fixed storage location at the time it is added to the database.

1 19. A method of maintaining an index during a reorganization of data in a
2 database, wherein said database comprises one or more records, each said record
3 includes a root segment and one or more non-root segments, and each root and non-
4 root segment comprises a prefix component and a data component, the method
5 comprising:

6 storing said prefix component of each root segment in a separate storage
7 location, wherein each storage location is associated with a particular
8 block of storage locations;

9 retaining said prefix component of each root segment in its storage location
10 during a reorganization of data in said database; and

11 storing said prefix component of each non-root segment which is associated
12 with a first root segment, in a block of storage locations in which said
13 prefix component of said first root segment is also stored.

1 20. The method of claim 19, wherein all root segments stored within a
2 block of storage locations are stored in contiguous storage locations.

1 21. The method of claim 19, wherein said database is an IMS full function
2 database.

1 22. A method of facilitating correction of an index after a reorganization of
2 data in a database, wherein said index comprises index entries, said database
3 comprises one or more records, each said record comprises one or more segments,
4 and each index entry comprises an address to a target segment, the method
5 comprising:

6 prior to a reorganization of data in said database, assigning a unique token to
7 each target segment and each corresponding index entry having an
8 address to a target segment, wherein said unique token for a given
9 target segment and for a corresponding index entry is the same;
10 after a reorganization of data in said database, reading the unique token of a
11 first index entry;
12 reading the unique token of each target segment until a match is found
13 between the unique token of a matching target segment and the unique
14 token of said first index entry;
15 determining the address of said matching target segment; and
16 replacing the address of said first index entry with the address of said
17 matching target segment.

1 23. The method of claim 19, wherein said index which is to be corrected is
2 a secondary index and each said segment is a non-root segment.

1 24. The method of claim 22, wherein said database is an IMS database.

1 25. The method of claim 23, wherein said database is an IMS database.

1 26. A method of facilitating correction of an index after a reorganization of
2 data in a database, wherein said index comprises index entries, said database
3 comprises one or more records, each said record comprises a root segment and one or
4 more non-root segments, and each index entry comprises an address to a target
5 segment included within said root and non-root segments, the method comprising:
6 prior to a reorganization of data in said database, assigning a unique token to
7 each target segment and each corresponding index entry having an
8 address to a target segment, wherein said unique token for a given
9 target segment and for a corresponding index entry is the same;
10 after a reorganization of data in said database, reading the unique token of a
11 first index entry for a first record;
12 reading the unique token of each non-root segment within said first record
13 until a match is found between the unique token of a matching target
14 segment and the unique token of said first index entry;
15 determining the address of said matching target segment; and
16 replacing the address of said first index entry with the address of said
17 matching target segment.

1 27. The method of claim 26, wherein said index which is to be corrected is
2 a secondary index.

1 28. The method of claim 26, wherein said database is an IMS database.

1 29. The method of claim 27, wherein said database is an IMS database.

1 30. The method of claim 26, wherein each said unique token includes one
2 or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token
4 is assigned.

1 31. A method of facilitating correction of an index after a reorganization of
2 data in a database, wherein said index comprises index entries, said database
3 comprises one or more records, each said record comprises a root segment and one or
4 more non-root segments, and each index entry comprises an address to a target
5 segment included within said root and non-root segments, wherein said root segment
6 and one or more non-root segments for a record are stored within a block of storage
7 locations, the method comprising:

8 prior to a reorganization of data in said database, assigning a unique token to
9 each target segment and each corresponding index entry having an
10 address to a target segment, wherein said unique token for a given
11 target segment and for a corresponding index entry is the same;
12 after a reorganization of data in said database, reading the unique token of a
13 first index entry for a first record;
14 reading the unique token of each non-root segment within said first record
15 until a match is found between the unique token of a matching target
16 segment and the unique token of said first index entry;
17 determining the address of said matching target segment; and
18 replacing the address of said first index entry with the address of said
19 matching target segment.

1 32. The method of claim 31, wherein said index which is to be corrected is
2 a secondary index.

1 33. The method of claim 31, wherein said database is an IMS database.

1 34. The method of claim 32, wherein said database is an IMS database.

1 35. The method of claim 31, wherein each said root segment is stored in a
2 fixed storage location prior to a reorganization of data in said database, and said root
3 segment is retained in said fixed storage location during a reorganization.

- 1 36. The method of claim 31, wherein each said unique token is includes
- 2 one or more the following: (i) a born on date of the target segment to which said
- 3 unique token is assigned; or (ii) a key field of the target segment to which said unique
- 4 token is assigned.

1 37. A method of facilitating correction of an index after a reorganization of
2 data in a database, wherein said index comprises index entries, said database
3 comprises one or more records, each said record comprises a root segment and one or
4 more non-root segments, each root and non-root segment comprises a prefix
5 component and a data component, and each index entry comprises an address to the
6 prefix component of a target segment included within said root and non-root
7 segments, the method comprising:

8 prior to a reorganization of data in said database, assigning a unique token to
9 the prefix component of each target segment and each corresponding
10 index entry having an address to the prefix component of a target
11 segment, wherein said unique token for the prefix component of a
12 given target segment and for a corresponding index entry is the same;

13 after a reorganization of data in said database, reading the unique token of a
14 first index entry for a first record;

15 reading the unique token of the prefix component of each non-root segment
16 within said first record until a match is found between the unique token
17 of a matching target segment prefix component and the unique token
18 of said first index entry;

19 determining the address of said matching target segment prefix component;

20 and

21 replacing the address of said first index entry with the address of said
22 matching target segment prefix component.

1 38. The method of claim 37, wherein said index which is to be corrected is
2 a secondary index.

1 39. The method of claim 37, wherein said database is an IMS database.

1 40. The method of claim 37, wherein each said root segment is stored in a
2 fixed storage location prior to a reorganization of data in said database, and said root
3 segment is retained in said fixed storage location during a reorganization.

1 41. The method of claim 37, wherein each index entry and each non-root
2 target segment further comprise a root segment identifier which identifies what root
3 segment said non-root target segment is associated with, and each identified root
4 segment comprises addresses to all non-root segments, within a record, associated
5 with said identified root segment.

1 42. The method of claim 41, wherein each said unique token includes one
2 or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token
4 is assigned.

1 43. The method of claim 42, wherein each said unique token for an index
2 entry and each non-root target segment further comprises said root segment identifier
3 which identifies what root segment said non-root target segment is associated with.

1 44. The method of claim 41, wherein said database is an IMS database.

1 45. The method of claim 41, wherein said index which is to be corrected is
2 a secondary index.

1 46. The method of claim 37, wherein each said unique token includes one
2 or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token
4 is assigned.

1 47. The method of claim 22, wherein after a reorganization of data in said
2 database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said
4 address only if it is invalid.

1 48. The method of claim 47, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index

3 entry to the unique token associated with a segment located at said address, and
4 ascertaining if said unique tokens are the same.

1 49. The method of claim 47, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry
3 to a segment code associated with a segment located at said address, and if said
4 segment codes are the same, then comparing the unique token of said first index entry
5 to the unique token associated with said segment located at said address, and
6 ascertaining if said unique tokens are the same.

1 50. The method of claim 26, wherein after a reorganization of data in said
2 database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said
4 address only if it is invalid.

1 51. The method of claim 50, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index
3 entry to the unique token associated with a segment located at said address, and
4 ascertaining if said unique tokens are the same.

1 52. The method of claim 50, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry
3 to a segment code associated with a segment located at said address, and if said
4 segment codes are the same, then comparing the unique token of said first index entry
5 to the unique token associated with said segment located at said address, and
6 ascertaining if said unique tokens are the same.

1 53. The method of claim 31, wherein after a reorganization of data in said
2 database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said
4 address only if it is invalid.

1 54. The method of claim 53, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index
3 entry to the unique token associated with a segment located at said address, and
4 ascertaining if said unique tokens are the same.

1 55. The method of claim 53, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry
3 to a segment code associated with a segment located at said address, and if said
4 segment codes are the same, then comparing the unique token of said first index entry
5 to the unique token associated with said segment located at said address, and
6 ascertaining if said unique tokens are the same.

1 56. The method of claim 37, wherein after a reorganization of data in said
2 database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said
4 address only if it is invalid.

1 57. The method of claim 56, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index
3 entry to the unique token associated with a prefix component of a segment located at
4 said address, and ascertaining if said unique tokens are the same.

1 58. The method of claim 56, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry
3 to a segment code associated with a prefix component of a segment located at said
4 address, and if said segment codes are the same, then comparing the unique token of
5 said first index entry to the unique token associated with said prefix component of a
6 segment located at said address, and ascertaining if said unique tokens are the same.

1 59. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 1.

1 60. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 7.

1 61. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 13.

1 62. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 19.

1 63. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 22.

1 64. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 26.

1 65. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 31.

1 66. A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 37.